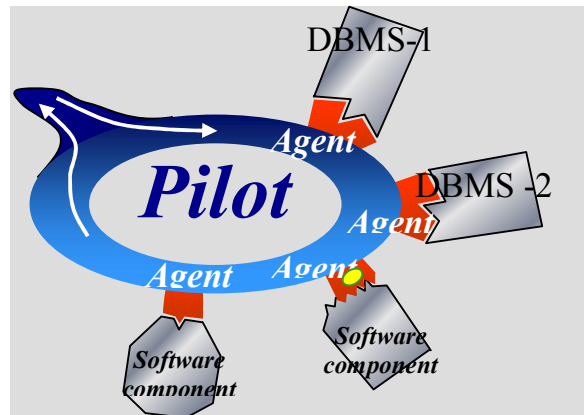


Pilot: a new generation middleware



Concepts

Pilot objectives - Rationale behind the Pilot

The Pilot is a middleware that eases the development and maintenance of applications that need to integrate various components, such as databases, mail servers, decision support systems or any pluggable software component.

The Pilot can greatly speed up the development of applications from simple ones to highly complex ones.

The Pilot has been designed to offer *Intelligent Brokering Facilities*. This means that it aims to facilitate the management of complex services. By complex service, we mean any service that interacts with different components or/and where business rules need to be implemented.

Target applications

The Pilot has been designed for target applications in the medical domain. It is the framework for hospital applications integrating distributed heterogeneous databases and decision support systems. Since then, the Pilot proved to be useful in many enterprise domains. It has for instance been used for:

- Software architecture of a medical network between in-town physicians, patients and hospitals
- Travel reservation
- Insurance
- Web site for financial information
- Knowledge management applications
- Traceability applications

It can also be used as a powerful engine for data migration in the context of a re-engineering process.

Model

The Pilot has a **Java** Application Programming Interface. It is then delivered as Java package on which you can build your own applications. It gives access to Components by the means of specific Agents and the access is controlled by rules described as Services.

Components

The **Components** are the different software elements that the application integrates: they may be relational databases, legacy systems, mail systems, decision support systems or any software.

Services

The Pilot manages **Services**. A service is a large-grain functionality (it can for instance be: select data, sort them, send the result by mail). A service is described in XML and is decomposed into simple elementary Steps. The Pilot manages the execution of the services. Roughly speaking, executing a service consists in executing the different executable steps. A step is considered executable if the conditions that can be associated to it are true. A step is where the link to a given component is realised.

Agents

The integration of software components is performed through the means of a special Java class called **class Agent**. You can build your own Agent classes or benefit from ready-to-use classes such as ones developed for relational databases or mail management.

Development process

The development of an application based on the Pilot first requires to integrate the Pilot into your own environment (this may be straightforward in case of a Java environment, a little more complex if you are in a C or C++ environment). You then need to develop your own Agent classes (if the ready-to-use do not cover your needs) and translate your business rules into XML Services.

Ready-to-use Agents

Powerful Agents are packaged into the Pilot framework. Among them, the **XMLJDBCAGENT** allows an easy integration of relational databases. This integration is based on query models described in XML. Then the integration of a specific database component can simply be performed by writing an XML query model, without writing any Java code.

Technology and position

The Pilot is a pure Java middleware. It relies on XML since:

1. XML is used as the Services description language.
2. The Document Object Model (or DOM), an XML-associated standard, which allows to view an XML document as a tree structure is used as the exchange format between the Pilot and the external components.

With the ready-to-use Agents, it is more than a middleware and can be seen as a performant development and execution framework.

The Pilot position is different from the position of famous standards of Distributed Computing, such as Corba or Web Services. If you need to access a Corba server object or a specific Web

Service, this will be done through a dedicated Agent: a Corba Client Agent or a Web Service Client Agent. Then, we can for instance imagine a complex service combining Corba, SQL databases and Web Services.

J2EE Integration

One of the most common use of the Pilot is for Web-based applications. So, the Pilot has been integrated into a J2EE platform where Services are executed by a generic servlet. It greatly speeds up the development and maintenance since the application can be highly configured thanks to XML. It then becomes a Web Development Framework named **Pilot Your Web**.

Miscellaneous

A scheduler has been developed (beta version) to perform batch processes: an XML configuration allows the description of the management of repetitive or delayed tasks modelled as Pilot services.

Two new domains of applications are being explored :

- In Peer-to-Peer applications based on the JXTA platform, services can be deployed over a Peer-to-Peer network.
- The XML Services description language is used to design grid applications where parallel executions can be performed over a grid.

Contact : sauquet@itssauquet.com